IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Serial No.:	09/330,852	Confirmation No.:	9904
Applicant:	David L. Reese, et al.		
Title:	PROFILING RANGES OF EXECUTION OF A COMPUTER PROGRAM		
Filed:	June 11, 1999		
Art Unit:	2193		
Examiner:	John Q. Chavis		
Atty. Docket:	114596-07-4014		
Customer No.	38492		

AFTER FINAL – EXPEDITED PROCEDURE

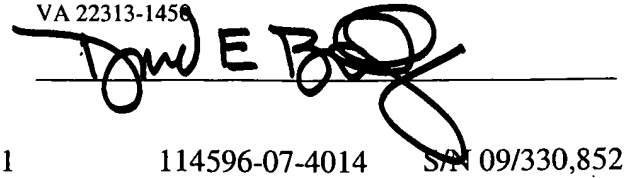
CONDITIONAL SUPPLEMENTARY AMENDMENT

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Applicant hereby responds to the Office Action of September 30, 2005. In the event that the accompanying Request to Withdraw Finality is granted, or else pursuant to 37 C.F.R. § 1.116(b)(2), kindly amend the application as follows.

AMENDMENTS TO THE CLAIMS begin on page 2 of this paper.

I certify that this correspondence, along with any documents referred to therein, is being deposited with the United States Postal Service on November 30, 2005 as First Class Mail in an envelope with sufficient postage addressed to Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.



AMENDMENTS TO THE CLAIMS

1. (previously presented) A method, comprising:

executing a program on a computer, without the program having been compiled for profiled execution, the program being coded in an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in the binary representation of the instruction, the computer including instruction pipeline circuitry configured to execute instructions of the computer, and profile circuitry configured to detect and record, without compiler assistance for execution profiling, profile information describing a sequence of events occurring in the instruction pipeline;

during a profile-quiescent interval of execution of the program that induces events that match time-independent selection criteria of profileable events to be profiled, configuring the profile circuitry to record no profile information in response to the occurrence of profileable events;

after a triggering event is detected, the triggering event being one of a predefined class of triggering events, configuring the profile circuitry to commence a profiled execution interval and to record in a memory of the computer profile information describing every event during a profiled execution interval that matches the time-independent profileable event selection criteria induced during the profiled execution interval, including at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution;

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change taken together with a processor mode before the mode change instruction the recording continuing until a predetermined stop condition is reached;

the recorded profile information stored in the memory of the computer and having a data structural form efficiently tailored to annotate the profiled binary code with sufficient processor mode information to resolve mode-dependency in the binary coding, and indicating contiguous ranges of sequential instructions executed during a profiled interval by low and high boundaries of the contiguous ranges, indicating the high boundary by the address of the last byte of a multi-byte instruction that ends the range, the profile information further identifying each distinct physical page of instruction text executed during the execution interval.

2. (previously presented) A method, comprising:

executing a program on a computer;

recording in a memory of the computer profile information concerning the execution of the program, the profile information recording the address of the last byte of at least one multi-byte instruction executed by the computer during a profiled interval of the execution.

3. (previously presented) The method of claim 2:

wherein the program has been compiled, without special consideration for execution profiling, into an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction;

and further comprising recording profile information describing the processor mode during the profiled execution interval, the profile information having a data structural form efficiently tailored to annotate the profiled binary code with sufficient processor mode information to resolve mode-dependency in the binary coding.

4. (original) The method of claim 2:

wherein the program has been compiled without special consideration for execution profiling;

and further comprising, commencing the profiled execution interval at the expiration of a timer, the recorded profile describing a sequence of events including every event that matches time-independent selection criteria of events to be profiled, the recording continuing until a predetermined stop condition is reached.

5. (original) The method of claim 2:

wherein the program has been compiled without special consideration for execution profiling;

wherein the program execution induces occurrence of events that match time-independent criteria of profileable events to be profiled;

and further comprising, during a profile-quiescent interval of execution, recording no profile information in response to the occurrence of profileable events; and

commencing the profiled execution interval after a triggering event is detected, the triggering event being one of a predefined class of triggering events, the recorded profile information describing every event that matches the profileable event selection criteria induced after the triggering event, the recording continuing until a predetermined stop condition is reached.

6. (original) The method of claim 5, wherein the triggering event is the expiration of a timer.

7. (original) The method of claim 5:

and further comprising, without software intervention, recording for later analysis a profile entry noting the source and destination of a control flow event in which control flow of the program execution diverges from sequential execution.

8. (original) The method of claim 5:

wherein the program is coded in an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction;

the recorded profile information describing at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution;

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change, the instruction opcode taken together with a processor mode before the mode change instruction;

the recorded profile information further identifying each distinct physical page of instruction text executed during the profiled execution interval.

9. (original) The method of claim 5:

wherein the program is coded in an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction;

and further comprising, recording profile information describing at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution, and

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change, the instruction opcode taken together with a processor mode before the mode change instruction.

10. (original) The method of claim 5:

the profile information further identifying each distinct physical page of instruction text executed during the profiled execution interval.

11. (original) The method of claim 5, wherein the program is executed on a computer having:

an instruction pipeline configured to execute instructions from a memory of the computer; and

profile circuitry configured to detect the occurrence of profileable events occurring in the instruction pipeline, and to direct the instruction pipeline to record profile information describing the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention.

12. (original) The method of claim 11, wherein:

the profile information is recorded into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

13. (original) The method of claim 5, wherein:

the profile information is recorded into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

14. (previously presentedl) The method of claim 13, the recorded profile information having a data structural form efficiently tailored to identify all bytes of object code executed during the profiled execution interval, without reference to the binary code of the program.

15. (original) The method of claim 2:

wherein the program has been compiled without special consideration for execution profiling;

and further comprising, without software intervention, recording for later analysis a profile entry noting the source and destination of a control flow event in which control flow of the program execution diverges from sequential execution.

16. (original) The method of claim 2:

wherein the program has been compiled, without special consideration for execution profiling, into an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction;

the recorded profile information describing at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution;

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change, the instruction opcode taken together with a processor mode before the mode change instruction;

the recorded profile information further identifying each distinct physical page of instruction text executed during the profiled execution interval.

17. (original) The method of claim 2:

wherein the program has been compiled, without special consideration for execution profiling, into an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction;

and further comprising, recording profile information describing at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution, and

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change, the instruction opcode taken together with a processor mode before the mode change instruction.

18. (original) The method of claim 17, wherein the divergence from sequential execution flow is consequent to recognizing or handling of an exception.

19. (original) The method of claim 17, wherein:

an instruction of the computer, having a primary effect on the execution the computer not related to profiling, has an immediate field for an event code encoding the nature of a profileable event to be recorded in the profile information, the immediate field having no effect on computer execution except to determine the event code of the profiled event.

20. (original) The method of claim 17, further comprising:

recording profile information that records a sequence of events of the program, the sequence including every event during the profiled execution interval that matches time-independent criteria of profileable events to be profiled.

21. (original) The method of claim 2:

wherein the program has been compiled, without special consideration for execution profiling;

the profile information further identifying each distinct physical page of instruction text executed during the profiled execution interval.

22. (previously presented) The method of claim 2, the recorded profile information having a data structural form efficiently tailored to identify all bytes of object code executed during the profiled execution interval, without reference to the binary code of the program.

23. (original) The method of claim 2, wherein the recorded physical memory references include addresses of binary instructions referenced by an instruction pointer, and at least one of

the recorded instruction references records the event of a sequential execution flow across a page boundary in the address space.

24. (previously presented) A computer, comprising:

an instruction pipeline configured to execute instructions of the computer;

profile circuitry configured to detect, without compiler assistance for execution profiling, the occurrence of profileable events occurring in the instruction pipeline, and to direct recording into the memory of the computer of profile information representing the events, the profile information recording of the address of the last byte of at least one multi-byte instruction executed by the computer during a profiled interval of the execution.

25. (previously presented) The computer of claim 24, wherein:

the instruction pipeline is configured to execute instructions of an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction; and

the profile circuitry is configured to record profile information describing a processor mode during the profiled execution interval, the profile information having a data structural form efficiently tailored to annotate the profiled binary code with sufficient processor mode information to resolve mode-dependency in the binary coding.

26. (previously presented) The computer of claims 25, the recorded profile information further having a data structural form efficiently tailored to identify all bytes of object code executed during the profiled execution interval, without reference to the binary text of the program.

27. (currently amended) The computer of claim 24, wherein:

~~during a profile-quiescent interval of execution of a program that has been compiled without special consideration for execution profiling and that induces events that match time-independent criteria of profileable events to be profiled,~~ the profile circuitry is configured to record no profile information in response to the occurrence of profileable events during a profile-quiescent interval of execution of a program that has been compiled without special

consideration for execution profiling and that induces events that match time-independent criteria of profileable events to be profiled; and

~~after a triggering event is detected, the triggering event being one of a predefined class of triggering events,~~ the profile circuitry is configured to commence the profiled execution interval after a triggering event is detected, the triggering event being one of a predefined class of triggering events, and to record profile information describing every event that matches the profileable event selection criteria induced during the profiled execution interval, the recording continuing until a predetermined stop condition is reached.

28. (original) The computer of claim 24:

the profile circuitry being configured to record at least one physical memory reference noting the source and destination of a control flow event in which control flow of the program execution diverges from sequential execution.

29. (original) The computer of claim 24, wherein:

the instruction pipeline is configured to execute instructions of an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction; and

the profile circuitry is configured to record profile information describing at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution, and

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change, the instruction opcode taken together with a processor mode before the mode change instruction.

30. (original) The computer of claim 29, wherein:

the profile circuitry is configured to detect the occurrence of profileable events occurring in the instruction pipeline, and to direct the instruction pipeline to record profile information describing the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention.

31. (original) The computer of claim 29, wherein:

the profile circuitry is configured to record profile information into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

32. (original) The computer of claim 24, wherein:

the profile circuitry is configured to record profile information identifying each distinct physical page of instruction text executed during the profiled execution interval.

33. (original) The computer of claim 24, wherein:

the instruction pipeline is configured to execute instructions of an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction; and

the profile circuitry is configured to record profile information describing at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution;

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change, the instruction opcode taken together with a processor mode before the mode change instruction; and

the profile circuitry is configured to record profile information identifying each distinct physical page of instruction text executed during the profiled execution interval.

34. (previously presented) A method, comprising:

executing a program on a computer, without the program having been compiled for profiled execution, the program being coded in an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in the binary representation of the instruction;

recording in a memory of the computer profile information describing an interval of the program's execution and processor mode during the profiled interval of the program, the profile

information having a data structural form efficiently tailored to annotate the profiled binary code with sufficient processor mode information to resolve mode-dependency in the binary coding.

35. (previously presented) The method of claim 34, the recorded profile information further having a data structural form efficiently tailored to identify all bytes of object code executed during the profiled execution interval, without reference to the binary code of the program.

36. (original) The method of claim 34:

wherein the program has been compiled without special consideration for execution profiling;

wherein the program execution induces occurrence of events that match time-independent criteria of profileable events to be profiled, the criteria including at least some physical memory references referenced by the program;

and further comprising, during a profile-quiescent interval of execution, recording no profile information in response to the occurrence of profileable events; and

commencing the profiled execution interval after a triggering event is detected, the triggering event being one of a predefined class of triggering events, the recorded profile information describing every event that matches the profileable event selection criteria induced after the triggering event, the recording continuing until a predetermined stop condition is reached.

37. (original) The method of claim 36, wherein the triggering event is the expiration of a timer.

38. (original) The method of claim 34:

wherein the program has been compiled without special consideration for execution profiling;

and further comprising, commencing the profiled execution interval at the expiration of a timer, the recorded profile describing a sequence of events including every event that matches

time-independent selection criteria of events to be profiled, the recording continuing until a predetermined stop condition is reached.

39. (original) The method of claim 34:

and further comprising, without software intervention, recording for later analysis a profile entry noting the source and destination of a control flow event in which control flow of the program execution diverges from sequential execution.

40. (original) The method of claim 34, wherein the program is executed on a computer having:

an instruction pipeline configured to execute instructions from a memory of the computer; and

profile circuitry configured to detect the occurrence of profileable events occurring in the instruction pipeline, and to direct the instruction pipeline to record profile information describing the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention.

41. (original) The method of claim 34, wherein the recorded physical memory references include addresses of binary instructions referenced by an instruction pointer, and at least one of the recorded instruction references records the event of a sequential execution flow across a page boundary in the address space.

42. (original) The method of claim 41, wherein the recorded profile information includes a record denoting a page boundary of the address space occurring within a single instruction.

43. (original) The method of claim 41, wherein the recorded profile information includes a record denoting a page boundary between two instructions that are sequentially adjacent in the logical address space.

44. (original) The method of claim 34, further comprising:

recording profile information recording a data-dependent change to a full/empty mask for registers of the computer.

45. (original) The method of claim 34, wherein a profile entry describing a single profileable event explicitly describes a page offset of the location of the event, and inherits a page number of the location of the event from the immediately preceding profile entry.

46. (previously presented) A computer, comprising:

an instruction pipeline configured to execute instructions of the computer, coded in an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in the binary representation of the instruction;

profile circuitry configured to detect, without compiler assistance for execution profiling, the occurrence of profileable events occurring in the instruction pipeline, and to direct recording in a memory of the computer profile information describing an interval of the program's execution and processor mode during a profiled interval of the program, the profile information having a data structural form efficiently tailored to annotate the profiled binary code with sufficient processor mode information to resolve mode-dependency in the binary coding.

47. (original) The computer of claim 46, wherein:

during a profile-quiescent interval of execution of a program that has been compiled without special consideration for execution profiling and that induces events that match time-independent criteria of profileable events to be profiled, the profile circuitry is configured to record no profile information in response to the occurrence of profileable events; and

after a triggering event is detected, the triggering event being one of a predefined class of triggering events, the profile circuitry is configured to commence the profiled execution interval and to record profile information describing every event that matches the profileable event selection criteria induced during the profiled execution interval, the recording continuing until a predetermined stop condition is reached.

48. (original) The computer of claim 46:

the profile circuitry being configured to record at least one physical memory reference noting the source and destination of a control flow event in which control flow of the program execution diverges from sequential execution.

49. (original) The computer of claim 46, wherein:

the instruction pipeline is configured to execute instructions of an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction; and

the profile circuitry is configured to record profile information describing at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution, and

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change, the instruction opcode taken together with a processor mode before the mode change instruction.

50. (original) The computer of claim 46, wherein:

the profile circuitry is configured to record profile information identifying each distinct physical page of instruction text executed during the profiled execution interval.

51. (original) The computer of claim 46, wherein:

the instruction pipeline is configured to execute instructions of an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction; and

the profile circuitry is configured to record profile information describing at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution;

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change, the instruction opcode taken together with a processor mode before the mode change instruction; and

the profile circuitry is configured to record profile information identifying each distinct physical page of instruction text executed during the profiled execution interval.

52. (previously presented) A method, comprising:

during a profiled interval of an execution of a program on a computer, recording in a memory of the computer profile information describing the execution, without the program having been compiled for profiled execution, the program being coded in an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in the binary representation of the instruction, the recorded profile information describing at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution;

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change taken together with a processor mode before the mode change instruction;

the profile information further identifying each distinct physical page of instruction text executed during the execution interval.

53. (previously presented) The method of claim 52:

and further comprising, without software intervention, recording for later analysis a profile entry noting the source and destination of a control flow event in which control flow of the program execution diverges from sequential execution.

54. (previously presented) The method of claim 52, wherein the program is executed on a computer having:

an instruction pipeline configured to execute instructions from a memory of the computer; and

profile circuitry configured to detect the occurrence of profileable events occurring in the instruction pipeline, and to direct the instruction pipeline to record profile information describing the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention.

55. (previously presented) The method of claim 54, wherein:

the profile information is recorded into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

56. (previously presented) The method of claim 54, the recorded profile information having a data structural form efficiently tailored for storage in the memory of the computer to identify all bytes of object code executed during the profiled execution interval, without reference to the binary code of the program.

57. (previously presented) The method of claim 54, wherein the recorded profile information describes a sequence of program events during the profiled interval of execution, the sequence including every event that matches time-independent criteria of profileable events to be profiled.

58. (previously presented) The method of claim 54, further comprising:

when an instruction fetch of an instruction causes a miss in a translation look aside buffer (TLB), the fetch of the instruction triggering a profileable event, servicing the TLB miss and reflecting the corrected state of the TLB in the profile information recorded for the profileable instruction.

59. (previously presented) The method of claim 52, further comprising the steps of:

executing the program on a first CPU of a multiprocessor computer;

on a second CPU of the multiprocessor, while the execution and profiling of the program continues, analyzing the collected profile data;

controlling the execution of the program on the first CPU based at least in part on the analysis of the collected profile data.

60. (previously presented) The method of claim 52, wherein:

the profile information is recorded into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

61. (previously presented) The method of claim 60, wherein the recorded profile information describes a sequence of program events during the profiled interval of execution, the sequence including every event that matches time-independent criteria of profileable events to be profiled.

62. (previously presented) The method of claim 52:

wherein the program has been compiled without special consideration for execution profiling;

wherein the program execution induces occurrence of events that match time-independent criteria of profileable events to be profiled, the criteria including at least some physical memory references referenced by the program;

and further comprising, during a profile-quiescent interval of execution, recording no profile information in response to the occurrence of profileable events; and

commencing the profiled execution interval after a triggering event is detected, the triggering event being one of a predefined class of triggering events, the recorded profile information describing every event that matches the profileable event selection criteria induced after the triggering event, the recording continuing until a predetermined stop condition is reached.

63. (previously presented) The method of claim 62, wherein the recorded profile information is efficiently tailored for storage in the memory of the computer to identify all bytes of object code executed during the profiled interval, without reference to the binary code of the program.

64. (previously presented) The method of claim 62, further comprising:

dividing the criteria for profileable events into initiating events and non-initiating events;

after the triggering event is detected, ignoring non-initiating profileable events; and

when an initiating event is detected, commencing recording the profile entries in the memory, describing every initiating and non-initiating event matching the profileable criteria during an interval following the triggering event.

65. (previously presented) The method of claim 52, further comprising:

commencing the profiled execution interval at the expiration of a timer, the recorded profile describing a sequence of events including every event that matches time-independent selection criteria of events to be profiled, the recording continuing until a predetermined stop condition is reached.

66. (previously presented) The method of claim 52, wherein the recorded physical memory references include addresses of binary instructions referenced by an instruction pointer, and at least one of the recorded instruction references records the event of a sequential execution flow across a page boundary in the address space.

67. (previously presented) The method of claim 15, wherein at least one of the recorded instruction references records the event of a page boundary of the address space occurring within a single instruction.

68. (previously presented) The method of claim 15, wherein the recorded profile information includes a record denoting sequential flow across a page boundary lying between two instructions that are sequentially adjacent in the logical address space..

69. (previously presented) The method of claim 52, further comprising:

recording profile information recording a data-dependent change to a full/empty mask for registers of the computer.

70. (previously presented) A computer, comprising:

an instruction pipeline configured to execute instructions of the computer, the program being coded in an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in the binary representation of the instruction;

profile circuitry configured to detect, without compiler assistance for execution profiling, the occurrence of profileable events occurring in the instruction pipeline, and to direct recording of profile information describing the detected profileable events, the recorded profile information describing at least all events occurring during a profiled execution interval of the two classes:

a divergence of execution from sequential execution;

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change taken together with a processor mode before the mode change instruction;

the profile information further identifying each distinct physical page of instruction text executed during the execution interval.

71. (previously presented) The computer of claim 70, wherein:

the profile circuitry is configured to detect the occurrence of profileable events occurring in the instruction pipeline, and to direct the instruction pipeline to record profile information describing the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention.

72. (previously presented) The computer of claim 71, wherein:

the profile circuitry is configured to record profile information into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

73. (previously presented) The computer of claim 70:

wherein the instruction pipeline and profile circuitry are components of a first CPU of a multiprocessor;

and further comprising a second CPU of the multiprocessor, configured to analyze the recorded profile information while the execution and profiling of the program continues on the first CPU, and to at least partially control the operation of the first CPU based at least in part on the analysis of the collected profile data.

74. (previously presented) The computer of claim 70, wherein:

the profile circuitry is configured to record profile information into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

75. (currently amended) The computer of claim 70, further comprising:

profile control bits implemented in the computer hardware, values of the profile control bits designed to control ~~controlling~~ a resolution of the operation of the profile circuitry;

a binary translator configured to translate programs coded in the instruction set in which an interpretation of an instruction depends on a processor mode not expressed in the binary representation of the instruction, being a first instruction set architecture, into instructions of a second instruction set architecture;

a profile analyzer configured to analyze the recorded profile information, and to set the profile control bits to values to improve the operation of the binary translator.

76. (previously presented) The computer of claim 70, wherein the profile circuitry is configured to record profile information efficiently tailored for storage in the memory of the computer to identify all bytes of object code executed during the profiled interval, without reference to the binary code of the program.

77. (previously presented) The computer of claim 70, wherein:

the profile circuitry is designed to record profile information that records a sequence of events of the program during the profiled execution interval, the sequence including every event that matches time-independent criteria of profileable events to be profiled.

78. (previously presented) The computer of claim 70, wherein:

the profile circuitry is designed based on a division of the profileable events into initiating events and non-initiating events;

triggering circuitry of the profile circuitry is designed to recognize a triggering event, and then to ignore non-initiating events in favor of one of the initiating events; and

the profile circuitry is designed to commence to record the profile entries in the memory, describing initiating and non-initiating events, after the initiating event is detected.

79. (previously presented) The computer of claim 70, wherein the profile circuitry includes a timer interval value, specifying a frequency at which the profile circuitry is to monitor the instruction pipeline for profileable events.

80. (previously presented) The computer of claim 70:

wherein the instruction pipeline is configured to execute instructions of two substantially disjoint instruction sets, a native instruction set providing access to substantially all of the resources of the computer, and a non-native instruction set providing access to a subset of the resources of the computer.

81. (previously presented) The computer of claim 70, wherein the instruction pipeline and profile circuitry are further configured to effect recording of profile information describing an interval of the execution of an operating system coded in the non-native instruction set.

REMARKS/ARGUMENTS

This paper is a supplementary response to the Office Action of September 30, 2005. The shortened statutory period runs through December 30, 2005. Accordingly, this response is timely.

Claims 1-81 are now pending, a total of 81 claims. Claims 1, 2, 24, 34, 46, 52 and 70 are independent. Of the independent claims, claims 2, 24, 34, and 46 have not been allowed over the art, however, the Office Action is also insufficient to raise any rejection of these claims.

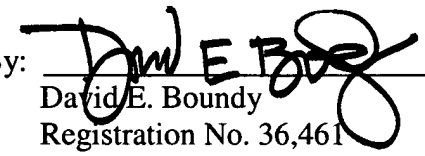
The amendments to the claims are not narrowing.

Applicant respectfully submits that the claims are in condition for allowance. Applicant requests that the application be passed to issue in due course. The Examiner is urged to telephone Applicant's undersigned counsel at the number noted below if it will advance the prosecution of this application, or with any suggestion to resolve any condition that would impede allowance. In the event that any extension of time is required, Applicant petitions for that extension of time required to make this response timely. Kindly charge any additional fee, or credit any surplus, to Deposit Account No. 23-2405, Order No. 114596-07-4014.

Respectfully submitted,

WILLKIE FARR & GALLAGHER LLP

Dated: November 30, 2005

By: 

David E. Boundy
Registration No. 36,461

WILLKIE FARR & GALLAGHER LLP
787 Seventh Ave.
New York, New York 10019
(212) 728-8757
(212) 728-9757 Fax